WEST

Generate Collection Print

L5: Entry 14 of 22 File: USPT Jan 4, 2000

DOCUMENT-IDENTIFIER: US 6012130 A
TITLE: Method and apparatus for automated disk drive upgrades

Abstract Text (1):
A computer executable program method for automating the upgrade of a storage medium, such as a disk drive, in a computer system is disclosed. The computer based method is preferably implemented with a series of computer-executable instructions stored on a computer-readable medium, such as a magnetic disk or CD-ROM. When executed on a computer, the program method automatically performs the series of operations needed to configure, partition and format the upgrade disk drive. The method also provides the option of adjusting system parameters so as to account for any changes to logical device designations resulting from the installation of the upgrade disk drive. Once installed, the program method optionally provides the user with the option of utilizing the upgrade disk drive as the new system boot device, or as a secondary/slave device. If the upgrade drive is to be used merely as a secondary drive, the program method can automatically transfer data and/or application files from the existing drive to the upgrade drive. If the upgrade drive is to be used as the new system boot device, the program method will automatically transfer operating system files from the preexisting drive to the upgrade drive, as well as any desired data and/or application files. Also, if the upgrade drive is to be used as the system boot drive, the method will provide the user with options for setting the appropriate system parameters. Thus, the upgrade drive can be designated as the boot drive by manipulating the appropriate jumper and cable settings, or the necessary changes can be invoked under the automatic control of the computer. Also, at the completion of the installation, the program method can be used to remove any transferred files from the preexisting drive, thereby freeing up storage space on that device.

Brief Summary Text (7):
This increase in demand for computer storage has coincided with an evolution in the development of hard disk drive technologies and products, which has resulted in the availability of new drives that are smaller, faster, and that have increased storage capacities. Moreover, these advances--like other areas of computer technologies--have been accompanied by a continued decrease in the price of disk drives. As would be expected, this continued demand on storage space, coupled with the decline in prices, has resulted in users frequently upgrading their desktop computer systems.

Brief Summary Text (9):
While upgrading an existing computer with a new disk drive(s) is often the more cost effective approach to increasing a user's available storage space, it is not without its drawbacks. Installation of a new drive often involves a series of time-consuming, error-prone steps. This is largely due to the fact that many of today's operating systems and software applications can not be simply "copied" from an existing disk, to a new upgrade disk, and then be operable. For instance, in a typical environment, the recommended procedure to follow when upgrading to a new disk drive is to first de-install the existing operating system and all existing software applications and then backing up all of the user and application data onto another media, such as tape or floppy disk. The new disk must then be physically installed and appropriately configured and formatted. The user must then reinstall and appropriately configure the operating system with the appropriate program and environment settings, and then reinstall each individual software application. Finally, the user and application data must be restored from the temporary media to the new disk.

Brief Summary Text (11):
Thus, what is needed is the ability to install a new disk drive in an existing computer system in a manner that is time efficient and not prone to error. Moreover, the process should be usable by computer novices, and should be capable of automatically

transferring and configuring the operating system, applications and data from the old drive to the new upgrade drive.

Brief Summary Text (13):
The foregoing problems in the prior state of the art have been successfully overcome by the present invention, which is directed to a computer program method and apparatus for automatically upgrading a disk drive on a desktop computer system. With the present invention, the user physically installs the upgrade disk drive into an existing computer, and then executes the present software based method, which automatically configures the upgrade disk drive media and then transfers the operating system(s), software applications and application data from the old drive to the new drive. This configuration and transfer of applications and data is accomplished in a manner such that the upgrade drive can subsequently function as the default boot drive for the computer system.

Brief Summary Text (15):
Since the present invention essentially automates the disk upgrade process, a user can quickly and easily upgrade an existing computer's disk storage capacity. The process eliminates many of the manual steps previously needed to perform an upgrade. Further, since the method automatically transfers the computer's existing operating system and applications, there is no need to perform a time consuming backup of the existing disk drive, and the user does not have to reinstall each computer software application to the new drive. Moreover, automation of the upgrade process eliminates any requirement that the user have detailed knowledge of complex operating system and hardware configuration procedures and commands, thereby allowing the upgrade to be performed by a novice computer user.

Detailed Description Text (3):
FIGS. 1 and 2, and the accompanying discussion, provide an overview of an exemplary computing environment in which the present invention may be implemented. Those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including without limitation multi-processor systems, network PCs, minicomputers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Detailed Description Text (14):
Once the new drive has been physically installed, the operating system and hardware must be appropriately configured so that the new drive is recognized and useable by the operating system and computer hardware. This particular configuration occurs at functional step 204. The exact series of processing steps performed here will likely depend on the particular operating system being used, and may also depend on the particular computer hardware environment. Thus, the preferred program method may include a menu containing options for selection by the user, so that the particular operating environment can be selected, and the correct operating sequence followed. Once selected, the program method will cause the computer's CPU 102 to perform the appropriate series of steps for identifying the newly connected disk drive. These series of steps can be hard coded by way of computer executable instructions, or the program method may simply invoke the underlying operating system utilities necessary to configure and set-up a new disk drive. For instance, in one embodiment, the program method may invoke the BIOS setup utility for the computer, and perform an autodetect function to identify the new drive and set the appropriate operating parameters for the drive.

Detailed Description Text (15):
Once the new drive has been detected, the computer CPU will continue processing in accordance with the program method of FIG. 3, and proceed to functional step 206. At that step, the series of computer-executable instructions needed to appropriately partition and format the new drive will be performed. Again, while this particular function could be carried out exclusively by including the requisite program instructions for the particular operating system/computer hardware, in the preferred embodiment the program method will utilize the underlying utilities already present in the particular operating system to perform the function. For instance, in one embodiment, the program method would invoke the utility known as "fdisk." The utility would be supplied with the appropriate parameters (either automatically or via appropriate prompts to the user) so as to select the new upgrade drive and partition it.

Detailed Description Text (16):
Once partitioned, in the preferred embodiment the appropriate operating system utility will be invoked so as to then format the new disk. By way of example, the "format" utility will be invoked to format the new drive, which may be designated at this point with the drive letter d:. Once formatted, the new upgrade drive will be capable of receiving and storing data.

Detailed Description Text (18):
In many operating systems the designation and assigning of a logical device identifier to the new drive, such as the drive letter d:, may affect the logical device designations previously assigned to other devices in the system. For instance, if prior to the installation of the upgrade disk drive the system was configured with a CD ROM optical disk drive, the logical device designation for that drive may be affected once the upgrade disk drive is installed. Prior to the installation of the upgrade disk drive, the CD ROM drive may have been designated as the d: drive. However, once the upgrade drive is installed and assigned that very same designation (i.e., the d: drive designation), many operating systems will automatically reassign the designation for the CD ROM drive, for instance, as the e: logical drive. Often, such a change would affect the operating and system parameters of the operating system and/or software applications, since there may be applications or system applications and/or utilities files that make reference to the CD ROM drive by way of its old designation, the lower case d: drive indicator. Of course, after the installation of the upgrade disk drive such a designation would no longer be accurate, resulting in various errors. As such, these inaccurate references to drive/device designations would need to be updated so that they correspond to the new designation for the affected devices, such as the CD ROM drive.

Detailed Description Text (20):
Once the upgrade disk drive has been appropriately formatted, and the device designations in the system have been appropriately modified, if needed, the set of computer-executable instructions shown at functional step 209 are executed. At this point, the user is presented with an option of maintaining the pre-existing disk drive as the primary boot drive (e.g., the c: drive), and use the upgrade disk as a secondary/slave disk drive (e.g., the d: drive). If so, there is no need to transfer any of the operating system files from the old drive to the new upgrade drive, and the CPU 102 proceeds with the execution of the computer executable instructions represented at functional step 210.

Detailed Description Text (21):
At this functional step, the user may be presented with the option of selecting particular data and/or application software that is to be transferred from the old drive to the new upgrade disk drive. In one embodiment, the user could specify exactly which files should be transferred to the new drive. Alternatively, a menu or listing could be displayed on the computer monitor 134 that lists each of the application programs and/or data files that exist on the old drive, thereby providing the ability for the user to specify which of those files should be transferred over to the new upgrade disk drive. Of course, the user would also have the option of not transferring any files to the new upgrade drive. Once the specified files are transferred to the new drive, functional step 228 is performed, which provides the user with the option of removing transferred files from the old drive so as to free up additional computer storage space. Once the appropriate transfer has taken place, the CPU 102 proceeds to the functional step indicated at 212, at which point the processing ends.

Detailed Description Text (23):
Once the operating system is copied to the upgrade drive, the processor performs the series of program instructions corresponding to the functional block shown at 216. That functional step is that portion of the program method that effects the copying of all the application software on the old drive to the new upgrade drive. In addition, any corresponding user data and files (e.g., word processing documents, spreadsheets, databases, etc.) are also copied to the new drive. Once again, in the preferred embodiment, the files are transferred by way of an appropriate operating system copying utility. While functional steps 214 and 216 are illustrated as comprising two different functional steps, in practice the steps could be combined and performed in one single copy operation. Alternatively, program step 216 could be implemented in a manner similar to that of program step 210, and only selected applications and/or data are transferred to this upgrade drive.

Detailed Description Text (24):
Processing then continues at functional step 218. This is the point in the program

method where the user selects the manner by which the new drive will be assigned boot drive responsibilities. In the illustrated embodiment, the user is presented with a menu having several options at step 218, and the processor will loop until an option is selected at functional program step 220. Once an option is selected, the CPU 102 will proceed with the program steps that correspond with the particular option selected, as is shown.

Detailed Description Text (25):
By way of example and not limitation, in the preferred embodiment the user will have two options with respect to the method used for the configuration of the upgrade drive as the boot disk. First, the user may choose to make the new upgraded disk the new boot disk, and the old disk the slave or secondary disk, by simply powering the computer down, and switching the appropriate cables between the two disks and changing the appropriate switch or jumper settings depending on the particular hardware system. Second, the user may choose to make the upgrade disk the new boot disk, and the old disk the slave or secondary disk, by way of software manipulation of the operating system settings (e.g., making the old drive the d: drive, and the new drive the c: drive), instead of making any physical changes to the system.

Detailed Description Text (26):
FIG. 3 illustrates examples of the functional program steps corresponding to each of the above options. Thus, if at functional step 220 the user chooses to make the upgrade drive the new boot disk, and the old drive the slave or secondary disk by switching the appropriate cables between the two disks and changing the appropriate switch settings, the processor will proceed to program step 222. At this step, the program method can optionally cause the computer display screen to display to the user the particular cable and switch setting information that must be made so as to successfully re-assign the particular boot drive settings. Of course, this information would vary depending on operating system, disk drive and computer system configuration types, and information for different manufacturers could be stored and then displayed based on equipment specifications entered by the user. The user would then power the system down and set the new upgrade drive to be the master boot drive (i.e., drive c:) by making the appropriate cable and switch settings and then either remove the old drive, or re-cable and set the appropriate switch settings so that it functions as a slave/secondary drive (i. e., drive d:). The computer system would then be in a condition to subsequently boot from the newly installed upgrade drive.

Detailed Description Text (27):
In the example embodiment illustrated in FIG. 3, the user also has the option of automatically designating the new drive as the default boot drive. Under this option, the program method manipulates the operating system parameters so as to logically re-map the addresses of the two disk drives so that the new drive, previously drive d:, will be mapped as the default boot drive (i.e., drive c:), and the old drive, previously drive c:, will be mapped as the slave/secondary drive (i.e., drive d:). By way of example, this functional step is represented at step 224 in FIG. 3. There, the computer processor executes a series of program steps so as to identify from the operating system the interrupts and the physical addresses of the old disk (drive c:) and the new disk (drive d:). At step 226, those physical addresses are re-mapped to virtual interrupt addresses in a manner so as to effectively switch the physical addresses of the two drives, i.e., the old c: drive is now viewed by the operating system as the d: drive and vice versa. Once the virtually remapping of the drives is complete, the physical locations of the drives has not changed, but the logical functioning of the drives has--the new disk will effectively function as the default boot drive c:, and the old disk will function as the slave/secondary drive d:. Under this particular approach, the installation of the upgrade drive, and its designation as the default boot drive, is completely automatic and performed under the control of the software. No hardware manipulation or alteration is required on the part of the user.

Detailed Description Text (28):
Once the upgrade disk has been appropriately configured as the primary boot drive, the user then has the option of freeing up storage space on the old drive by removing all of the files that have been transferred from that drive to the new drive. This step is accomplished at program step 228, which can automatically remove all of the transferred files from the old drive or, alternatively, the user can specify exactly which of the files should be erased. In this way, the space on the old drive can be freed up for use as new data/application storage. Alternatively, the old drive can be completely physically removed from the computer system if the space is no longer needed.

Detailed Description Text (30):

In summary, the above program method provides an easy-to-use computer program that automates the disk upgrade process. Instead of having to go through a series of manual, tedious and often error-prone operations, a user can simply install the new drive, and then execute the upgrade program to automatically transfer all system, application and data files to the new drive. Further, the program provides the user with several options with respect to the selection and configuration of the default boot drive, including an option of automatically designating the new drive as the boot drive. The program saves time, eliminates errors, and requires no specialized expertise on the part of the user.

CLAIMS:

4. A method for programmably installing an upgrade storage medium as defined in claim 3, wherein the copying of the operating system to the upgrade storage medium is performed in a manner so as to preserve all file and directory attributes of operating system files copied.

WEST

Print Selection

Help | Clear | Cancel | Print | Print First Page

Select?	Document ID	Section(s)	Page(s)	# Pages to print	Database
☑	6128734	all	all	15	USPT,PGPB,JPAB,EPAB,DWPI
☑	6012130	all	all	10	USPT,PGPB,JPAB,EPAB,DWPI
☑	6006029	all	all	25	USPT,PGPB,JPAB,EPAB,DWPI
☑	5983012	all	all	25	USPT,PGPB,JPAB,EPAB,DWPI
☑	5913037	all	all	16	USPT,PGPB,JPAB,EPAB,DWPI
☑	5907679	all	all	13	USPT,PGPB,JPAB,EPAB,DWPI
☑	5890214	all	all	18	USPT,PGPB,JPAB,EPAB,DWPI

Building cpk2 **Room** 4c32 **Printer** gbfrptr

Main Menu | Logout

WEST Search History

DATE: Friday, December 13, 2002

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L5	L4 and link$	22	L5
L4	L1 and (new near drive$)	28	L4
L3	L1 and ((add$) near drive$)	56	L3
L2	L1 and ((add$ or insert$) near drive$)	78	L2
L1	system near upgrad$	2302	L1

END OF SEARCH HISTORY

WEST

Generate Collection Print

L5: Entry 13 of 22 File: USPT Oct 3, 2000

DOCUMENT-IDENTIFIER: US 6128734 A
TITLE: Installing operating systems changes on a computer system

Abstract Text (1):
A computer system is upgraded while the computer system is functioning. The computer system has a first boot device with a first operating system and a second device. The method includes: preparing the second device within the computer system as a bootable device while the computer system is functioning under control of the first operating system; preparing the second device within the computer system to receive a second operating system while the computer system is functioning under control of the first operating system; loading the second operating system onto the second device while the computer system is functioning under control of the first operating system; and, rebooting the computer system such that the computer system is under control of the second operating system on the second device.

Brief Summary Text (7):
High availability computer systems must be able to restart as soon as possible after a system failure or upgrade. Frequent system upgrades or patches are often necessitated by the discovery of system errors or by the establishment of new system requirements. Unfortunately, operating system upgrades often require that the computer system be powered down or turned off. Also, computer system down time can occur as a result of an error or a failure during system setup during the upgrading procedure.

Brief Summary Text (8):
Commonly, when a computer system requires an upgrade, patches are made to the operating system of the computer system. Often, new patches are added to the patches from a previous system upgrade resulting in a large number of patches for each successive upgrade. The successful patches are then updated to other computer systems by replacing the present operating system of those computer systems with the new operating system including the successful patches. The new operating system is often installed from a storage device such as a tape drive to directly replace the present operating system needing the upgrade. Alternatively, the new operating system can be downloaded over a network.

Brief Summary Text (10):
backup of the present operating system usually must be made, and the upgrades or patches must be loaded while the computer system is down. Additionally, upgrades must often be made for multiple computer systems, causing down-time for all the systems involved. For example, in a network of servers, each server must be upgraded individually. That is, the system manager must power down a server, thereby limiting server availability by preventing user access, make a backup of the present operating system, load the new operating system, and test the new operating system. The system manager then repeats these actions for each of the other servers. The computer system being upgraded is usually down throughout the procedure, possibly for a period of hours. The cost for system downtime during these upgrades can be significant. Also, the chances for error or failure can be significant with this practice.

Brief Summary Text (12):
It has been discovered that an operating system may be upgraded by selecting and preparing a new boot device while the computer system is functioning, loading the new operating system onto the boot device while the computer system is functioning, and rebooting the computer system. Such a procedure advantageously allows the target system to be upgraded while minimizing the target system's resulting down time to that required for a target system reboot. Additionally, multiple target systems may be upgraded while minimizing the opportunities for operator error during each target system upgrade. Thus, the reliability and efficiency of making upgrades is improved

with a simplified upgrade procedure that also reduces costs resulting from system upgrades.

Brief Summary Text (13):
In one embodiment of the present invention, a computer system is upgraded while the computer system is functioning. The computer system has a first boot device with a first operating system and a second device. The method includes: preparing the second device within the computer system as a bootable device while the computer system is functioning under control of the first operating system; preparing the second device within the computer system to receive a second operating system while the computer system is functioning under control of the first operating system; loading the second operating system onto the second device while the computer system is functioning under control of the first operating system; and, rebooting the computer system such that the computer system is under control of the second operating system on the second device.

Brief Summary Text (14):
In another embodiment of the present invention, a computer system is upgraded while the computer system is functioning. The computer system has a first boot device with a first UNIX operating system and a second device. The method includes: preparing the second device within the computer system as a bootable device while the computer system is functioning under control of the first UNIX operating system; preparing the second device within the computer system to receive a second UNIX operating system while the computer system is functioning under control of the first UNIX operating system; loading the second UNIX operating system onto the second device while the computer system is functioning under control of the first UNIX operating system; and, rebooting the computer system such that the computer system is under control of the second UNIX operating system.

Drawing Description Text (6):
FIG. 4 shows a flow chart for preparing the alternative boot drive for the new operating system according to an embodiment of the present invention.

Detailed Description Text (4):
In the preferred embodiment, a target computer system is upgraded by selecting an alternate boot device attached to the target computer system and preparing it to boot the target computer system. An operating system from an already upgraded model computer system is loaded onto the alternate boot device of the target system. The upgrade becomes effective when the target system is rebooted using the alternate boot device.

Detailed Description Text (63):
Control then transitions to modify step 630. During modify step 630, any configuration files for the new system are modified if required. For example, in the preferred embodiment, the name of the system must be changed from that of model system 110 to that of model system 120 by modifying the file /mnt/etc/src. sh which defines the system name. Also, the following files are modified in the preferred embodiment: the file /mnt/etc/netlinkrc ensures that the appropriate network links are enabled; the file /mnt/etc/hosts adds the system internet protocol (IP) address; the file /mnt/etc/defaultrouter adds the router IP address; the file /mnt/etc/defaulthostname changes the host's name; the files /mnt/etc/defaultmask and /mnt/etc/defaultbroadcast define the class of addresses (network specific links); and, the files /mnt/etc/issue and /mnt/etc/motd optionally provide information to the system users.

Detailed Description Text (78):
The machine specific hardware address of the new drive is thus identified. The upgrade procedure is has prepared target computer system 120 to be rebooted with the new boot disk drive 124.

Detailed Description Text (89):
The procedure set forth above is then repeated for target computer systems 130, 140, and 150 so that all computer systems are upgraded with the new operating system.

CLAIMS:

1. In a first computer system responding to users under control of a first operating system, a method for upgrading the first computer system while the first computer system continues to function under control of the first operating system, the first computer system having a first boot device storing the first operating system and a second device, the method comprising:

preparing the second device within the first computer system as a bootable device while the first computer system is functioning under control of the first operating system;

preparing the second device within the first computer system to receive a second operating system while the first computer system is functioning under control of the first operating system;

loading the second operating system onto the second device while the first computer system is functioning under control of the first operating system; and

rebooting the first computer system such that the first computer system is functioning under control of the second operating system on the second device, the first operating system being unmodified during the first computer system upgrading, wherein the first computer systems continues to respond to users during the upgrading.

5. The method of claim 1 further comprising:

upgrading the first operating system on a second computer to form the second operating system, wherein the upgrading is performed before the loading of the second operating system onto the second device.